Parallelization of Decision Graph Bayesian Optimization Algorithm

G. Surya Datta*, A. Rakesh, N. Vijayasundhar
SCSE VIT UNIVERSITY
India

Abstract— The traditional Bayesian optimization algorithm (BOA) is used to generate optimal solutions in Bayesian networks. To enhance performance of BOA, Decision Graphs were introduced which are helpful in parameter (variable) saving. When number of nodes in Bayesian network increases, execution time for computing optimal solution also increases proportionally. So, this paper proposes Parallelization of Decision Graph Bayesian Optimization Algorithm (PDBOA) to reduce this high execution time. OpenMP compiler directives are used to perform this parallelization. Contains nodes with the variables from the modelled data set in a directed acyclic graph manner.

Keywords— Bayesian Network, Decision Graph, Optimization, OpenMP, Parallelization

I. INTRODUCTION

A directed acyclic graph containing nodes which are corresponding to variables in data set is Bayesian Network. To find optimal solution in typical conditions for these networks, Bayesian optimization algorithm is used. DBOA (Decision Graph Bayesian Optimization Algorithm) is used to reduce abundance (complexity) in conditional probability of the large network taken by Bayesian optimization algorithm. This DBOA is more robust in generating optimal solution and also has a speed relative merit when compared with other optimization algorithms. This paper proposes the technique for parallelization of existing Decision Graph Bayesian Optimization Algorithm, a) Set problem and population sizes for generation of optimal child population b) The algorithm builds the decision graph for each variable by dividing and merging nodes in the graph c) A set of k candidates are taken and the best candidate is placed with selected candidates as in the process of Tournament Selection method with replacement d) The replacement of worst solutions present in the population is done with their child solutions, when all worse solutions in original set are replaced and size of child solution is set to cent percent e) Bayesian Drichlet method[5] doesn’t store prior information such as high standard solutions and information about network but it can store maximum number of in-degree of nodes. f) The criteria for termination can be taken as max number of generations, evaluation of fitness, population containing max proportion of optima, when bits in all position are similar and optimum solution is found g) Set parallel constructs for the desired sections of algorithm h) Best child solution, Population bias, model for runtime child generation and reduced execution time are obtained.

II. RELATED WORK

The tomography reconstruction algorithms [2] uses Bayesian network for optimizing the functional variables efficiently. The functional substitution which has been used broadly in simplifying the steps of iterative process applies the Bayesian Optimization. When the number of steps in the iterative process increases, the running time for construction of Bayesian network increases. A dynamic Bayesian network based optimization framework [5] that is reliable along with the genetic algorithm has been introduced. The design parameter for such framework is system reliability and component dependency. The results of simulation shows that integrating the dynamic Bayesian network and genetic algorithm provides an efficient, powerful, reliable system designed component but executing such integrated approach sequentially consumes high execution time. In order to improve the efficiency of evolutionary computation while applying to practical problems, (Tabu-BOA) Tabu-Bayesian Optimization Algorithm [1] has been applied on decision problems for cabling and scheduling the operations in electric power plant. These hybrid evolutionary algorithms with the other competitive genetic algorithms and heuristics have many complex combinatorial problems. Thus it was very hard to solve via traditional programming and mathematical techniques sequentially.

The way of design for Bayesian Optimization Algorithm (BOA) parallelization [6] for solving typical optimization problems on nVidia graphics hardware with Compute Unified Device Architecture (CUDA) is gBOA (BOA on GPU). In best case one can get a faster speedup of up to thirteen times (13x). But, this algorithm is platform dependent, uses the software which is hardware compatible. Bayesian networks are used to model gene interactions in bioinformatics [4]. For solving big-scale problems, structure of Bayesian Network is Trained, which is a NP-hard problem and making that important to involve heuristics. Markov boundary of each node is computed for parents and child set. Using parallel computers Computation of parents and children sets is done till it is sufficient to infer all edges in the Bayesian network. But, directionality of the edges is not inferred from which comprehensive network cannot be trained. BSPEA, replaces crossover operators in traditional EAs by learning Bayesian Networks, The algorithm has a strong robustness and is capable of handling the better Pareto front faster, and. But, as the network grows the interaction between variables is not
much stronger [3]. Executing the algorithm parallel involves the concurrent computation of the processes or the threads at the same time, whereas programming in sequential manner involves the successive and ordered execution of the processes.

The figure 1 shows that the main program is divided into many parallel regions called sections that are executed simultaneously. It is also shown that the parallel regions can be further sub-divided into sections leading to nested parallelism. Thus we can conclude that whenever the number of parallel sections increases in the program the rate of execution also increases proportionally.

III. DATA FLOW DIAGRAM

![Data Flow Diagram](image)

The PDBOA has step by step implementation as shown in the figure 2.

1) All of first, the input sets are taken from the network 2) Setting of problem, population size is to be done in the way that as bigger as the problem size, the bigger population and longer run tasks so problem size can be bigger and population size should be as less as possible as the bigger population will lead to have heavy consumption of memory and slows down the process 3) the decision graph with encoding of many sets of constraints is constructed which is extension of decision tree which takes an advantage of having multiple parents to each non root node. . The construction of decision graph involves in several steps like (i) division of nodes and merging them that in turn updates the process for simple evaluation (ii) tournament selection with replacement is a process which enables to authenticate the pressure of selection with the dynamic variable size is done (iii) replacement of worst cases with child solutions is done with the specification of offspring number in percentage of population which is replaced with the worst cases in the solution (iv) Setting the termination criteria is done when the desired optimum solution with homogeneous positions of all the bits are converged i.e., generations and fitness evaluations are maximum. As the decision graph construction undergoes complex steps, parallelization is done to reduce the execution time and 4) finally; the best child solution is obtained with the reduction in execution time.
IV. IMPLEMENTATION

In this paper the Parallel Decision graph Bayesian Optimization Algorithm (PDBOA) is implemented using in c++ programming language on visual studio 2012, OpenMP. PDBOA mainly, parts are divided, and each part is executed as a section. The different parts in the section are graph, population, replace, getting the Bayesian optimization parameters. In the graph section the definition of the classes like Oriented Graph and Acyclic oriented graph are stored. These classes explain about modifications to be done in the graphs and their orientation. The population section has some modification with population pattern(set of characters) and the original pattern(set of characters). The replace section in the algorithm has generation of some random number. The swap function uses the change of population strings. The random number generated is used in the swap function such that the new child solution formed is with change of digits and the final pattern thus obtained. The Bayesian optimization parameters are initialized and the build of the bayes network is maintained in this section. The decision graph is build by using the constraints for building decisions. The statistics for minimum population and maximum population is computed with the available population. The divided parts in the algorithm are given as different section constructs in open mp execution. At beginning the sequential execution of the algorithm is performed and the execution time of the algorithm is noted. Now after parallelizing the algorithm the execution time is noted for different multicore processors from 2-core, 4-core, 8-core, 16-core, 32-core. These execution times are noted and then compared with corresponding sequential execution time. And thus found that execution time is much more lesser when parallel execution compared to sequential execution.

V. RESULTS AND ANALYSIS

In this paper, the algorithm for the construction of Bayesian networks using decision graph takes different sets of input with variable problem size and population size. The graph in the figure shows the execution of the Decision graph Bayesian optimization algorithm done in sequential and parallel manner. When the algorithm is executed sequentially, the execution time varies with small amount of time only, even though the number of cores was increased. Whereas after the introduction of the OpenMP compiler directives at the entry of each structured blocks in the code the main function spawns the child threads. Each section of the program has been executed parallel as different threads. This in turn decreases the overall execution time of the algorithm, also when the number of cores in the processor is increased there is a drastic reduction in the execution time achieving greater efficiency.

The figure shows the table containing the execution times of the algorithm when executed both sequentially and parallel. It clearly depicts that there is a drastic reduction in the execution time when the code with parallel constructs are executed using multi core processors.

<table>
<thead>
<tr>
<th>Mode Of Execution</th>
<th>Number Of Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Sequential Execution(ms)</td>
<td>60.12</td>
</tr>
<tr>
<td>Parallel Execution(ms)</td>
<td>60.25</td>
</tr>
</tbody>
</table>

Figure 3. Table for comparison of sequential and parallel execution

![Figure 4. Graph showing Sequential and parallel execution time in different cores](image-url)
VI. CONCLUSION

A keen analysis of sequential Decision graph Bayesian optimization algorithm is done and later the parallelized mechanism for the algorithm is proposed. The use of C++ language integrated with OpenMP compiler directives helps the algorithm to be parallelized. Though the population size and sample size increases the execution time of the algorithm is consistent. As the number of cores increases in the execution time of the algorithm decreases proportionally. Execution times for sequential and parallel algorithms are compared among different cores and are observed that algorithm for parallelization has shown much more efficient execution time.

ACKNOWLEDGMENT

Thankful to School Of Computer Science Engineering, VIT University for allowing to carry out this research work and also for providing moral support and infrastructures.

REFERENCES

[2]. Dingzhou Cao, Shaobai Kan and Cuny Yu Sun 'Design of Reliable System Based on Dynamic Bayesian Networks and Genetic Algorithm' in the proceedings of IEEE pg. 1-6, 2012.
[5]. Olga Nikolova and Srinivas Aluru 'Parallel Discovery of Direct Causal Relations and Markov Boundaries with Applications to Gene Networks' in the proceedings of International Conference on Parallel Processing pg. 512-521, 2011.